# **Introduction to PERL Programming**

## **What is Perl?**

Perl is a programming language especially designed for text editing. It is now widely used for a variety of purposes including Linux system administration, network programming, web development etc.

Perl is of great importance in a Linux operating system where it can be used to create programs, handle Databases and e-mails, GUI (Graphical User Interface) development, Networking and System Administration.

## **PERL V/s Shell Scripting**



Even though, shell scripting is available to programmers, they prefer [Perl](http://www.guru99.com/perl-tutorials.html) because:

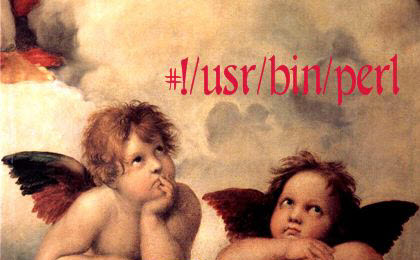
* Programming on  [Perl](http://www.guru99.com/perl-tutorials.html) does not cause portability issues, which is common when using different shells in shell scripting.
* Error handling is very easy on Perl
* You can write long and complex programs on [Perl](http://www.guru99.com/perl-tutorials.html) easily due to its vastness. This is in contrast with Shell that does not support namespaces , modules , object , inheritance etc.
* Shell has fewer reusable libraries available . Nothing compared to Perl's CPAN
* Shell is less secure. Its calls external functions(commands like mv , cp etc depend on the shell being used) . On the contrary [Perl](http://www.guru99.com/perl-tutorials.html) does useful work while using internal functions.

## **Perl Basics**

Always start your script with

**#!/usr/bin/perl**

It directs the execution to [Perl](http://www.guru99.com/perl-tutorials.html) interpreter on your system.



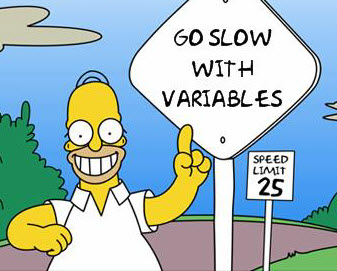
The path is usually the same on most of the Linux distributions.

## **Storing Variables, Input and Output**

|  |  |  |  |
| --- | --- | --- | --- |
| **Action** | **Description** | **Syntax** | **Example** |
| **Defining a Variable value** | Storing values to a Variable in form of string and number | **$variable = "value";** | **$name = "Ronald";** |
| **Output in Perl** | If you want a string or a value to display on the screen then you can use the print command | **print ("value to be printed") ;** | **Print("thanks");** |
| **Input in Perl** | If you want a user input to be assigned to a variable use | **$variable = ;** | **$username = ;** |

**Important points**

* With this, if you want the [Perl](http://www.guru99.com/perl-tutorials.html) interpreter to **ignore a statement, prefix it with a # symbol**.
* Remember that every **statement** in [Perl](http://www.guru99.com/perl-tutorials.html) ends with a **semi-colon**.
* **Perl is case-sensitive** . Make sure you use the right case.
* You can use any text editor to write your PERL scripts.
* You should then save the script **file in .pl extension** which will make it recognizable.
* Make sure you do not use spaces when you are naming the Perl script file.



## **Creating a PERL Script**

Let us understand the steps in creating a PERL Script

1. **Create a file** **using** a **vi** editor(or any other editor). Name script file with **extension .pl**
2. **Start** the script with **#! /bin/perl**
3. Write some code.
4. Save the script file as filename.pl
5. For **executing** the script type **perl filename.pl**

Let's write a PERL script which will take input from the user and display it back through the script.

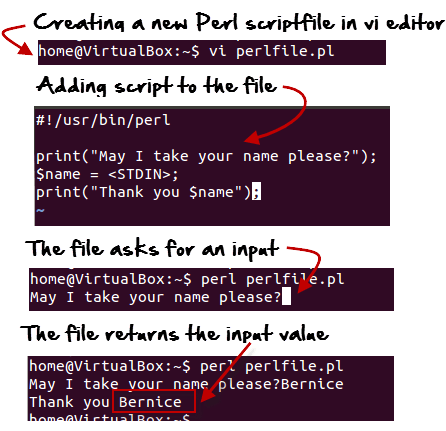
**#!/usr/bin/perl**

**print("May I take your name please?") ;**

**$name = ;**

**print("Thank you $name");**

Let's see the steps to create this script -



**Summary:**

* Perl is a general-purpose programming language originally developed for text manipulation
* Now used for a wide range of tasks including system administration, web development, network programming, GUI development, and more.
* Perl files have .pl extension
* There are three types of variables in Perl, Scalar, Lists and Hashes.